**Conocimiento Nº 5: AWT- Java Foundation Class (JFC)**

JFC, Java™ Foundation Class es un conjunto de componentes y características para ayudar a construir los entornos gráficos de los programas o GUIs (Graphical User Interfaces). Incluye prácticamente todo tipo de elementos gráficos:

* Soporte Look and Feel
* API de Accesibilidad
* Java 2D API (Java 2 Platform only)
* Soporte de Drag & Drop
* Swing

**Soporte Look and Feel**

Java utiliza el interfaz gráfico de la plataforma sobre la que se está ejecutando para presentar los Componentes del AWT con el aspecto asociado a esa plataforma, de este modo los programas que se ejecuten en Windows tendrán esa apariencia y los que se ejecuten en Linux tendrán otra apariencia. A su vez Swing permite la selección de esta apariencia gráfica, independientemente de la plataforma en que se esté ejecutando; tanto es así que, la apariencia por defecto de los Componentes Swing se denomina Metal, y es propia de Java

Cada ejecutable Java tiene un objeto **UIManager** que determina *el Look-and-Feel*, es decir, la *apariencia* en pantalla y *funcionamiento*, que van a tener los Componentes Swing de ese ejecutable. El *Look-and-Feel* es la apariencia que se proporciona a los diferentes Componentes: botones, cajas de texto, cajas de selección, listas, etc.

**API de Accesibilidad**

Por accesibilidad se entiende que las personas con algún tipo de discapacidad puedan participar de las actividades cotidianas de la vida, incluyendo el trabajo, el uso de servicios y productos, y el acceso a la información.

En el contexto informático las herramientas de accesibilidad son aquellas que pretenden facilitar el uso de las nuevas tecnologías en general a las personas con alguna discapacidad.

Java permite tecnologías asistidas como lectores de pantalla y display Braille para obtener información desde el interfase de usuario.

**Java 2D API**

Permite a los desarrolladores incorporar fácilmente gráficos 2D de alta calidad, texto, e imágenes en aplicaciones y applets Java.

**Soporte de Drag and Drop**

Proporciona la habilidad de arrastrar y soltar entre aplicaciones Java y aplicaciones nativas.

**Migración desde AWT**

* Swing permite la construcción de Interfaz gráfica para aplicaciones de cierta entidad
* Los componentes Swing son ligeros, en su mayoría, ya que no están basados en sus semejantes nativos como es el caso de AWT
* Swing incorpora un mayor número de componentes que AWT: tablas, árboles, sliders, barras de progreso, frames internos y componentes de texto.
* Swing permite unir eventos de teclado a componentes
* Para diferenciar los componentes Swing de los AWT, sus nombres están precedidos por una letra “J”.
* No se deben mezclar componentes Swing con componentes AWT en una misma aplicación
* Para poder migrar de AWT a Swing se debe considerar los siguiente:
  + Eliminar los import de paquetes de java.awt.\*
  + Importar el paquete **javax.swing.\***
  + Cambiar cada componente AWT por el Swing más parecido, por ejemplo:
    - Frame 🡪 JFrame
    - Button 🡪 JButton
  + Swing provee múltiples componentes de GUI que no están presentes en AWT, como Toolbar, ColorChooser, Table, Tree, DialogBoxes.

Tabla: En esta tabla están las clases de AWT representadas en Swing.

|  |  |
| --- | --- |
| Button | JButton |
| Canvas | JPanel |
| Checkbox | JCheckBox |
| Checkbox in CheckboxGroup | JRadioButton in ButtonGroup |
| Choice | JComboBox |
| Container | JPanel |
| List | JList |
| Choice | JComboBox |
| Container | JPanel |
| Component | JComponent |
| Label | JLabel |
| Menu | JMenu |
| MenuBar | JMenuBar |
| MenuItem | JMenuItem |
| Panel | JPanel |
| PopupMenu | JPopupMenu |
| Scrollbar | JScrollBar |
| TextField | JTextField |
| TextArea | JTextArea |
| ScrollPane | JScrollPane |
| Applet | JApplet |
| Dialog | JDialog |
| FileDialog | JFileChooser |
| Frame | JFrame |
| Window | JWindow |

# Modelo Vista Controlador

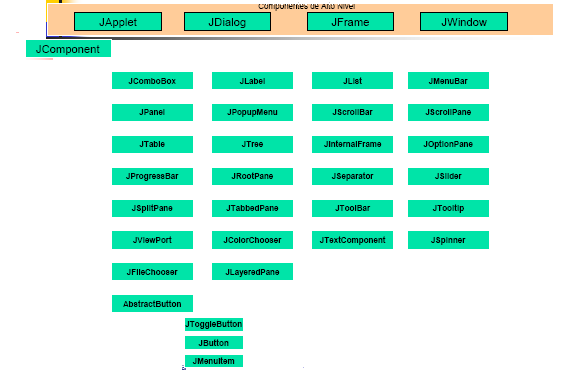
Corresponde a un patrón de diseño utilizado por Swing, del que se basa en el principio de la Programación Orientada al Objeto, donde las clases son diseñadas en base a responsabilidades, por lo que al programar un elemento gráfico, debemos considerar las siguientes clases:

* El Modelo, en el cual se almacena el contenido
* La Vista, que muestra el contenido
* El Controlador, que maneja las entradas del usuario

El patrón especifica la forma en que interactúan estos tres objetos.

**Componentes de Swing**

El siguiente esquema representa los componentes de la interfaz Swing:



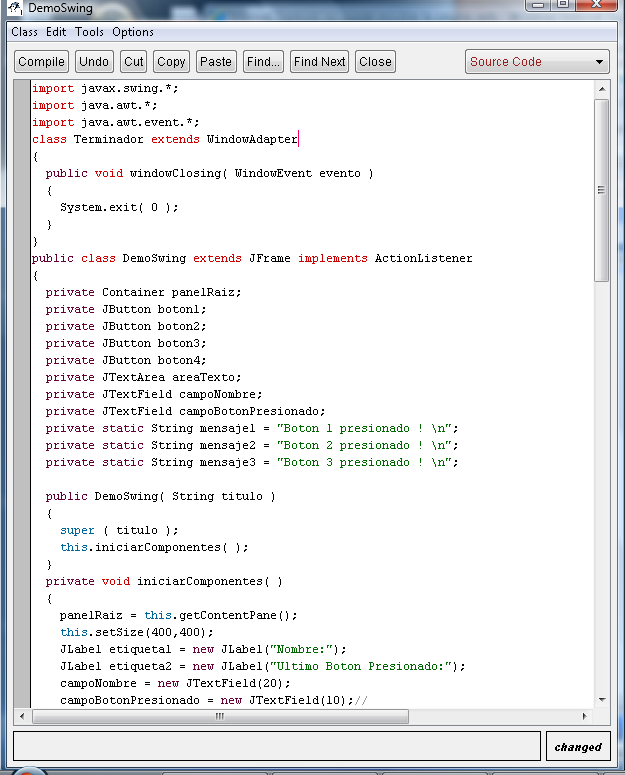
Resumen de Contenedores de Swing

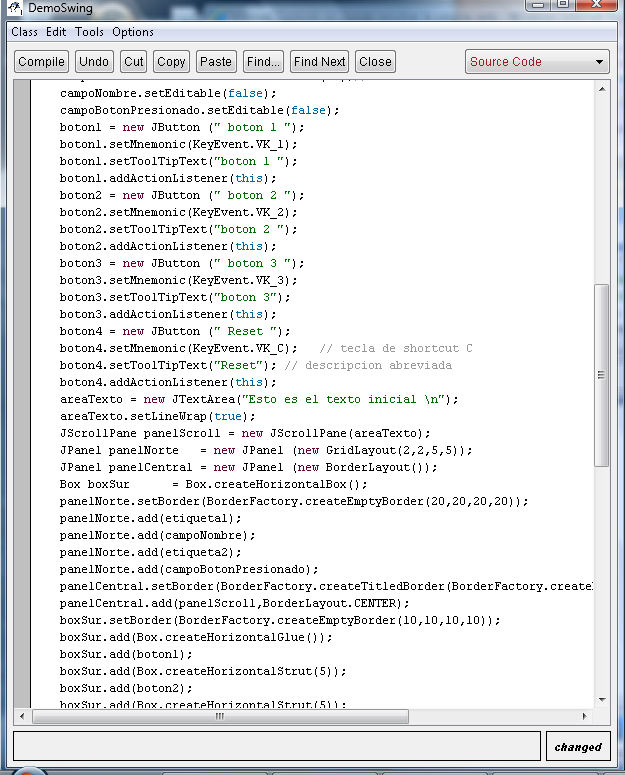
|  |  |
| --- | --- |
| **Contenedor** | **Descripción** |
| Box | Posiciona hijos usando BoxLayout. |
| JApplet | Applets. |
| JDesktopPane | Desktop que contiene JInternalFrame(s). |
| JDialog | Presentación de diálogos. |
| JFrame | Ventana. |
| JInternalFrame | Ventana interna. Suele usarse dentro de un JDesktopPane. |
| JLayeredPane | Contenedores apilados. |
| JPanel | Agrupar hijos. |
| JRootPane | Usado por JApplet, JDialog, JFrame, JInternalFrame y JWindow. Proporciona muchas características. |
| JScrollPane | Añade barras de desplazamiento a su hijo. |
| JSplitPane | Muestra dos hijos pudiendo ajustar sus tamaños relativos. |
| JTabbedPane | Solapas para mostrar diferentes hijos. |
| JViewPort | Muestra una parte de sus hijo. Típicamente usado por JScrollPane. |
| JWindow | Ventana sin decoración. |

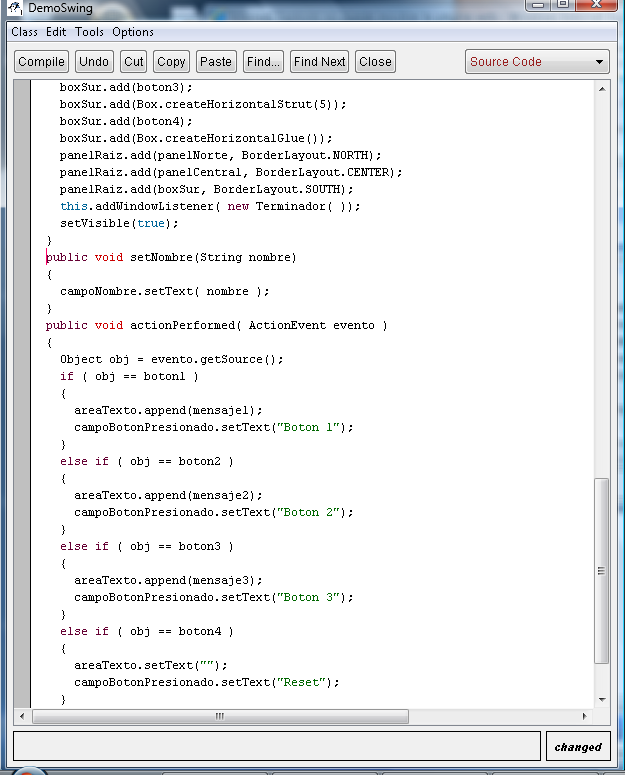
|  |  |
| --- | --- |
| **Componente** | **Descripción** |
| JButton | Botón. |
| JCheckBox | Botón de comprobación. |
| JCheckBoxMenuItem | Botón de comprobación para usar en menús |
| JColorChooser | Selector de colores. |
| JComboBox | Entrada de texto con lista de valores. |
| JComponent | Raíz de la jerarquía de componentes Swing. |
| JEditorPane | Editor de texto. Normalmente HTML o RTF. |
| JFileChooser | Selector de ficheros. |
| JLabel | Etiqueta. |
| JList | Lista. |
| JMenu | Menú dentro de un JMenuBar o dentro de otro menú. |
| JMenuBar | Barra de Menús. |
| JMenuItem | Elemento seleccionable en un menú. |
| JOptionPane | Ventanas de dialogo. |
| JPasswordField | Entrada de passwords. |
| JPopupMenu | Ventana con un menú. |
| JProgressBar | Barra de progreso. |
| JRadioButton | Botón excluyente. |
| JRadioButtonMenuItem | Botón excluyente para usar en menús |
| JScrollBar | Barra de desplazamiento. |
| JSeparator | Líneas de separación. |
| JSlider | Deslizador. |
| JTable | Tabla. |
| JTextArea | Edición de múltiples líneas de texto plano. |
| JTextComponent | Raíz de los editores de texto. |
| JTextField | Edición de una línea de texto plano. |
| JTextPane | Subclase de JEditorPane para hacer procesadores de texto. |
| JToggleButton | Padre de JCheckBox y JRadioButton. |
| JToolBar | Barra de herramientas o acciones. |
| JToolTip | Ventana informativa. |
| JTree | Árboles. |

Resumen de los componentes de la interfaz Swing

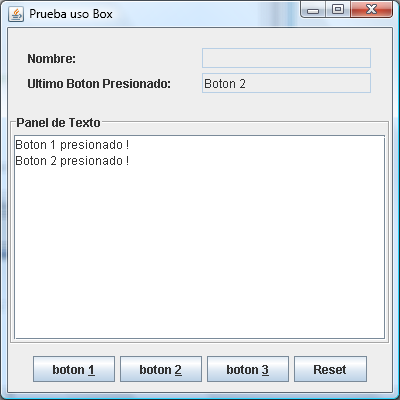
Ejemplo de uso de Box dentro de un contenedor de Swing



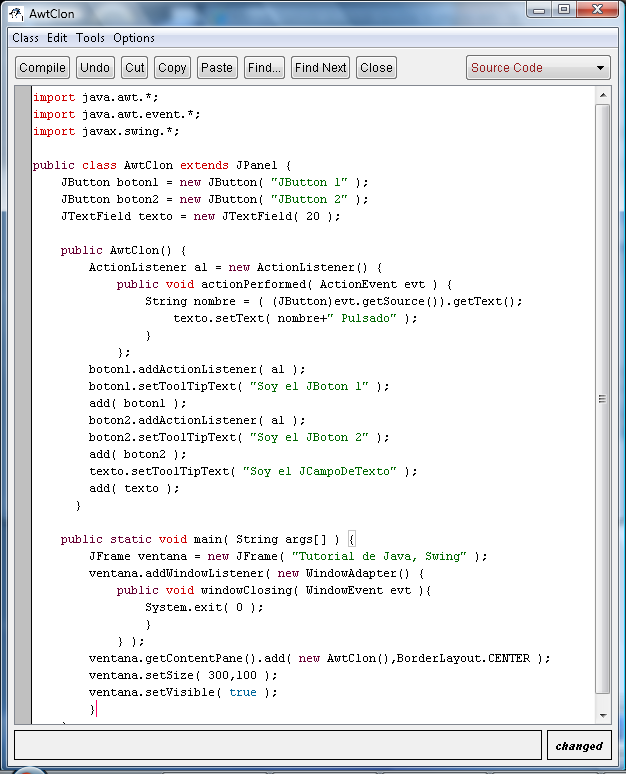




El resultado de ejecutar el código anterior es:



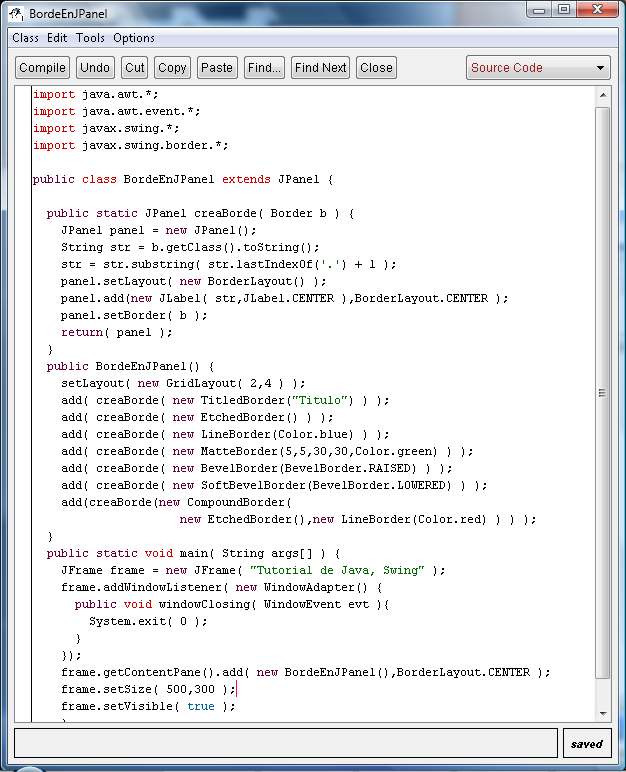
Ejemplo de uso de botones y campos de texto en Swing



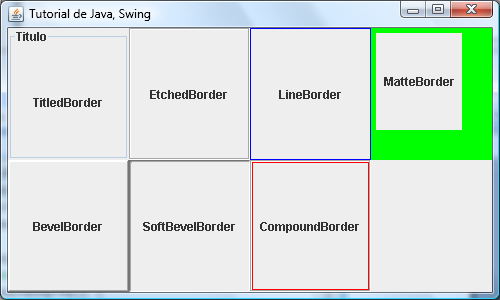
El resultado de la ejecución de este código es:



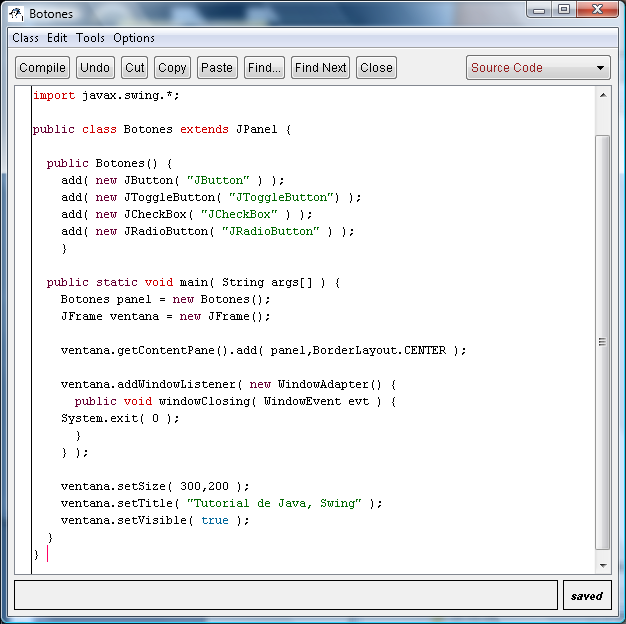
Ejemplo de uso de Bordes en los componentes gráficos:



El resultado de ejecutar el código anterior

:

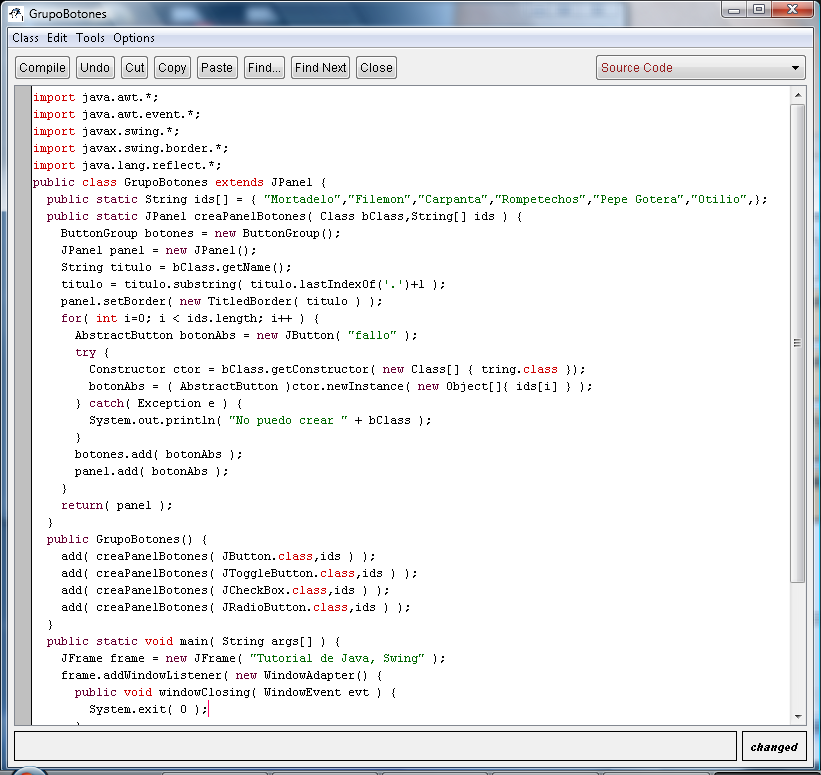
Ejemplo de uso de botones en Swing:

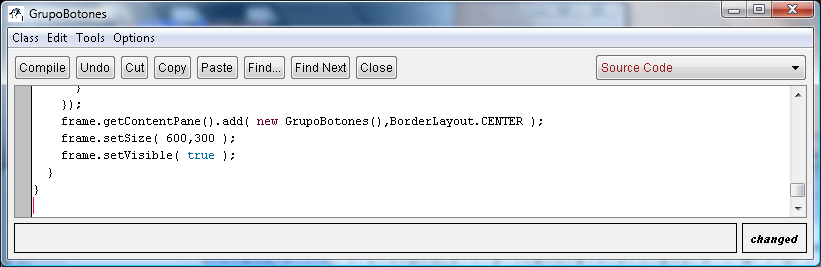


El resultado de ejecutar el código anterior es:



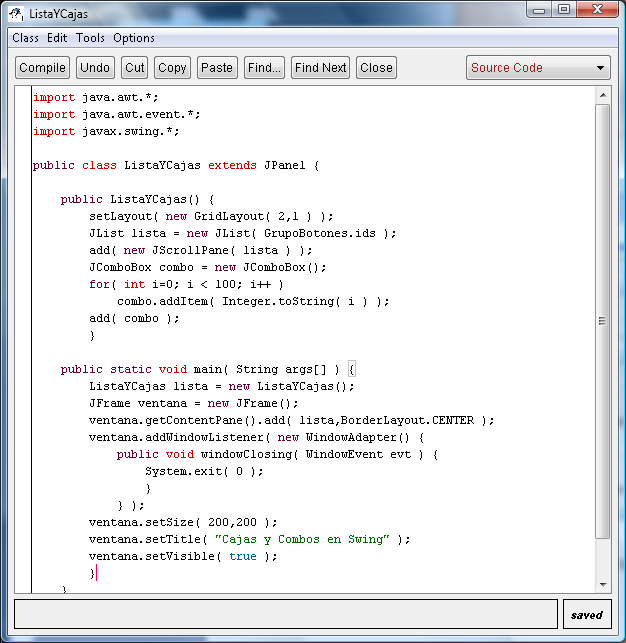
Ejemplo de código que muestra los distintos tipos de grupos de botones:



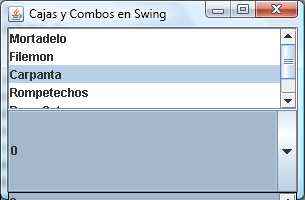
El resultado de ejecutar este código es:



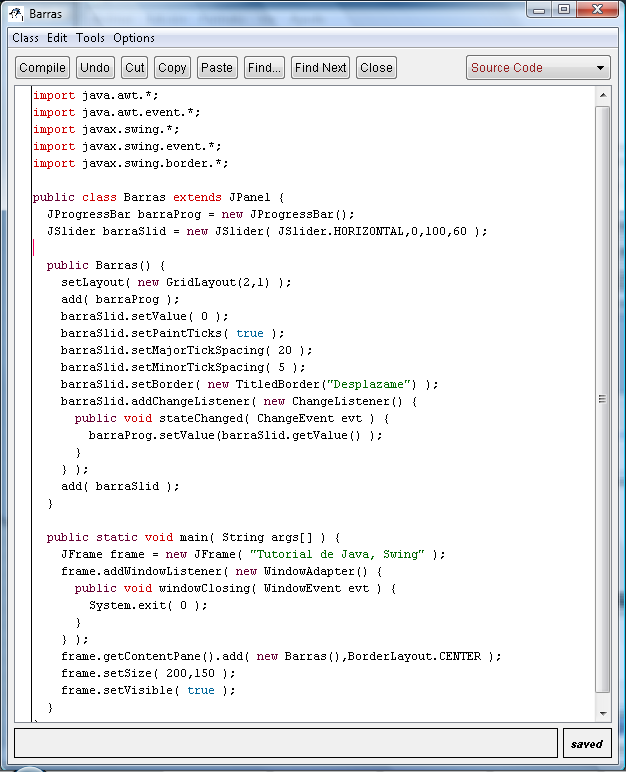
Ejemplo de uso de cajas y combos de listas:



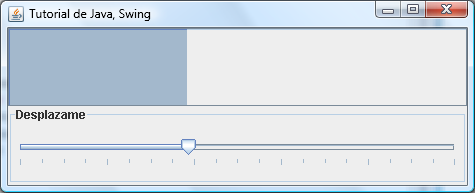
El resultado de ejecutar el código anterior es:

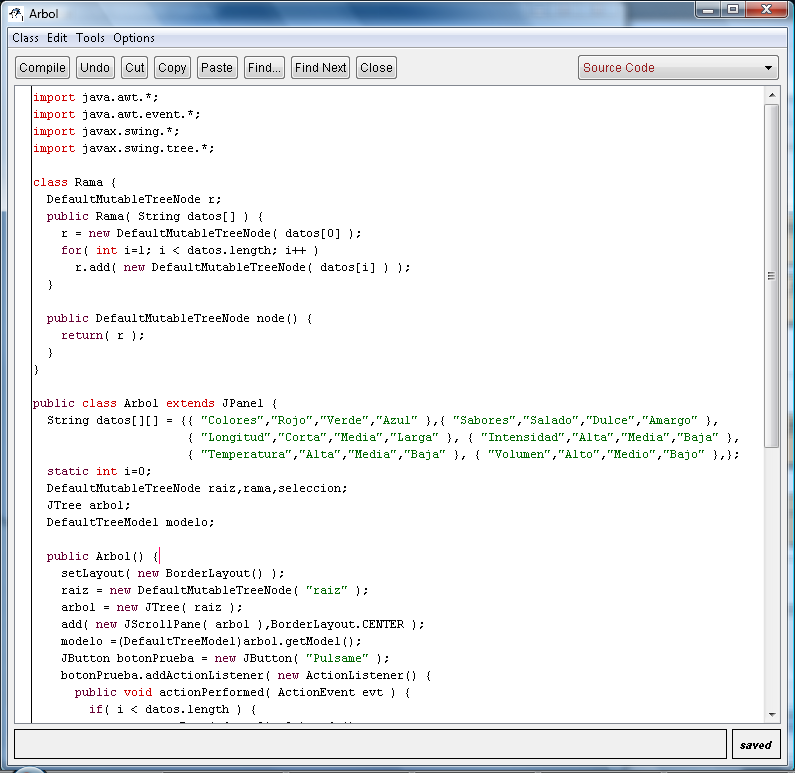


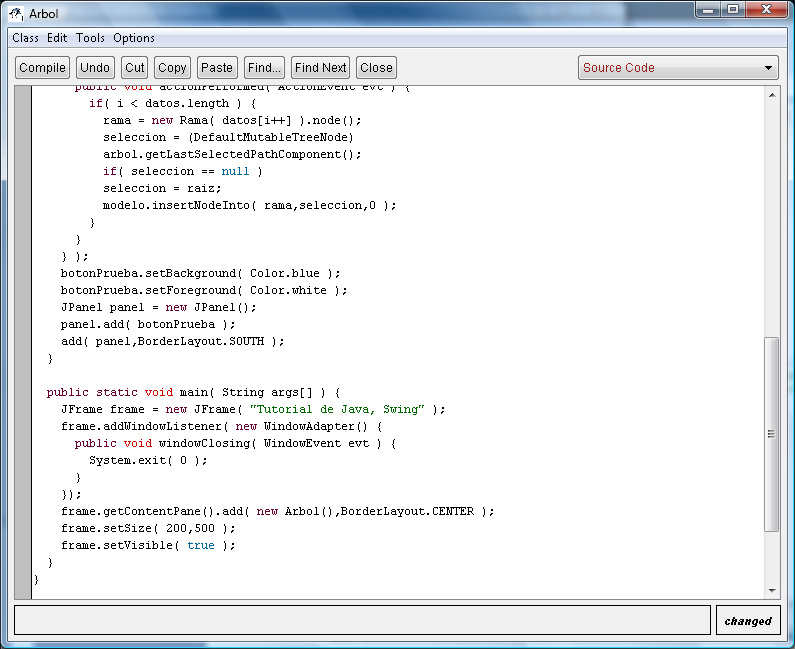
Ejemplo de uso de barras de desplazamiento



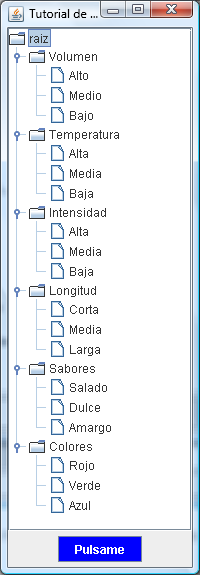
El resultado de ejecutar el código anterior es:

Ejemplo de uso de un Tree (árbol)

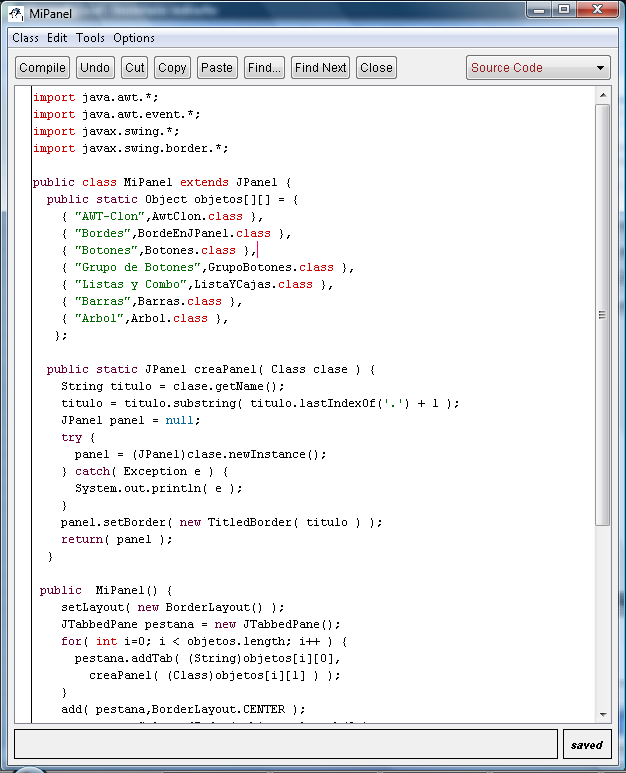


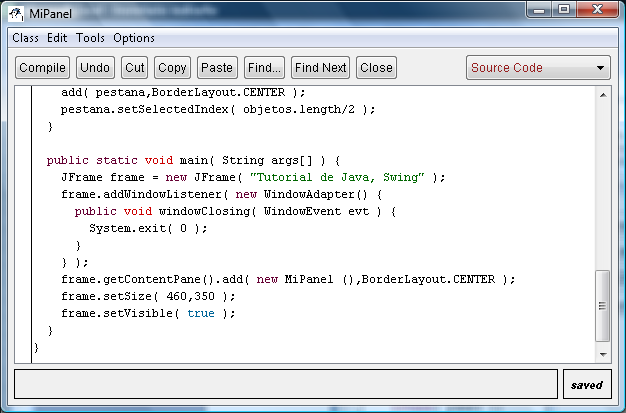


El resultado de ejecutar este código es:



Ejemplo de una aplicación que une todos los ejemplos anteriores en diferentes pestañas:





El resultado de ejecutar esta aplicación es:

